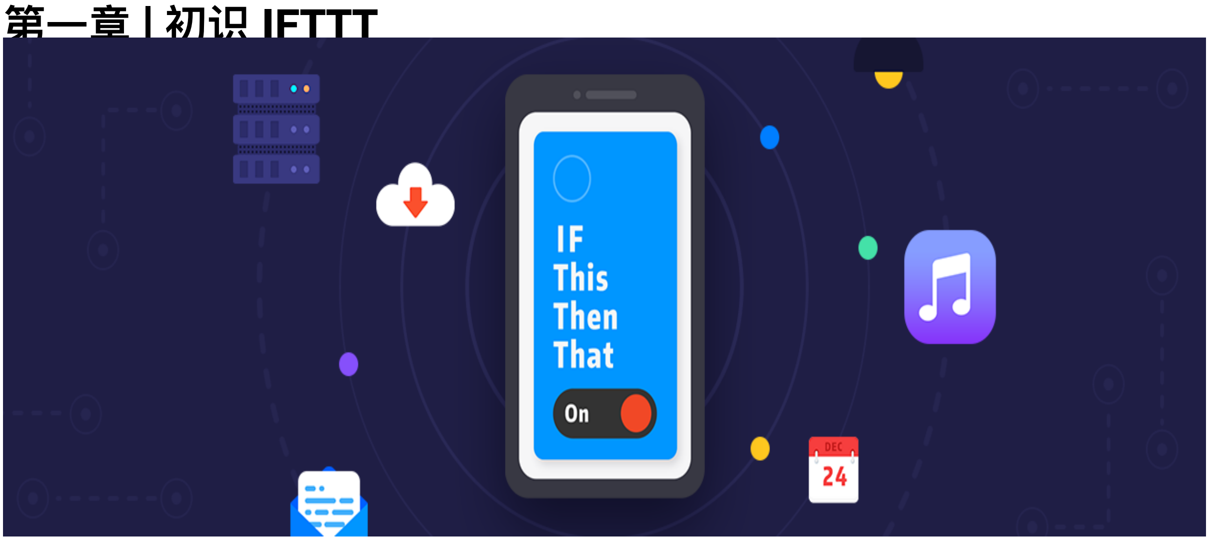
第一章 | 初识 IFTTT



王掌柜

03月28日

第一章 | 初识 IFTTT

| 本文为付费栏目文章，您已订阅，可阅读全文 |

欢迎大家阅读《玩转 IFTTT，互联网自动化也可以很简单：初识篇》，在系列文章中，我会由浅入深地告诉大家，什么是 IFTTT，如何使用 IFTTT 创建自动化任务，以及 IFTTT 都有哪些高阶玩法，只需要跟着系列文章顺序阅读，相信你也会成为一个人人羡慕的效率高手。

在这之前，我们先了解一下为什么会产生 IFTTT 这样的服务?

01. 为什么会出现 IFTTT

应用和服务的信息孤岛现象

如今社会中，互联网正在通过云计算、大数据、人工智能从方方面面渗透到我们每个人的工作和生活中，无论你从事什么职业，过着怎样的生活，很难想象没有互联网是怎样的日子。

但是互联网如此发达的今天，在我们的智能终端上，依然有很多应用和服务是各自独立存在的。有的应用只是提供阅读服务，有的应用只是提供音乐服务，这些应用可能很优秀，但是由于他们的各自独立存在，导致应用和服务之间不能互联互通，这在大数据和人工智能发展迅猛的今天带来了许多不便，这个现象我们称之为「信息孤岛现象」。大家通过下边例子理解一下：

事例 1：网易云音乐和 Apple Music 的信息孤岛现象

网易云音乐和 Apple Music 是两个非常优秀的流媒体音乐服务商，他们都有非常忠诚的拥趸，但是由于版权问题，许多用户选择同时使用两个服务。但是，这里有一个非常严重的问题，「无法在两个平台同步歌单」。这意味着你想在 Apple Music 听到网易云音乐歌单就必须手动再创建一遍，非常麻烦，这就是一个典型的**信息孤岛现象**。

我们对比看一下另一家优秀的流媒体音乐服务商 Spotify，Spotify 通过对外提供统一友好的 API，从而打破了**信息孤岛现象**，使得第三方应用可以对接到 Spotify 的流媒体服务，真正的做到了是音乐无处不在，典型的应用场景就是 Amazon 的 Echo，早上起床我们只要喊一声「Alexa, play Discover Weekly」，美妙的音乐就会从 Echo 响起来。

打破应用和应用之间壁垒

随着人们对生产力的需求提升，越来越多的人意识到单一的服务或应用是无法满足大部分需求的，必须要多个应用或服务协作才可以完成复杂的任务。打破服务和应用之间的壁垒，使得单一的服务或者应用不再形成孤岛，除了本身要提供统一的服务和支持，更重要的是对外要提供标准的输入和输出，与其他应用或者服务进行交互。为什么说这一点非常重要呢？我们从以下两点分析一下：

- 打破应用和应用之间的壁垒
- 打破服务和服务之间的壁垒

让我们把时间的指针回拨到「桌面应用时代」，那时候人们无论是电子邮件、通讯交流还是应用写作、制作报表等全部在本地完成，每一个应用都能独立完成自己提供的功能，但是几乎很少的应用考虑和别的应用交互，即使是数据导入导出，也因为没有标准格式，导致大部分应用之间不能通讯，这就是典型的**应用信息孤岛**。

随着互联网的发展，在 Mac 和 PC 上的应用通过各种方式尝试打破应用之间的壁垒，涉及到的技术包括不限于以下几种：

- AppleScript[1]
- Windows 批处理[2]
- 通用导出格式 HTML[3]、XML[4]、CSV[5]、Json[6]

这几种技术其实就两个方案：

1. 第一个方案，通过一些脚本语言，实现两个应用之间的沟通。
    - 缺点：需要一定的技术能力，许多人望而却步。

    - **例子 1：**我们可以通过 Windows 批处理，将 100 个 csv 格式文件转换成 xls 格式文件，不需要启动两个程序，只需要执行一段脚本即可。
    - **例子 2：**在前云时代[7]，我想通过 Hazel 将照片自动导入 DayOne 中生成一篇日记，就必须得借助 AppleScript 实现，大家可以阅读我以前写的《照片存储方案》了解。
2. 第二个方案，通过指定格式的导入导出功能，实现数据的互通。
    - 缺点：格式不统一，无法大规模推广和使用。
    - **例子：**如果想将为知笔记中的笔记导入到 Evernote 中，我们需要打开 PC 版为知笔记，将笔记导出成 HTML 格式，再打开 PC 版 Evernote，将刚才的 HTML 笔记导入即可。

十年前 Apple 公司发行了第一代 iPhone，一举揭开了移动互联网时代，新的时代也催生了新的技术，作为移动时代的领军平台 iOS 和 Android 也都遇到了「桌面时代」同样的问题**如何打破应用（app）之间的壁垒?**

iOS 平台打破 App 之间壁垒

iOS 作为一个目前最受欢迎的移动平台，没有独立的文件系统，应用之间不能互通这两个问题一直受人诟病。随着 iOS 系统的升级发展，Apple 也尝试许多新的技术来推动解决这个问题，包括新的存储系统，新的文件系统等，当然还有不得不提的就是 **URL Scheme** 协议。

关联阅读：《URL Schemes 使用详解》

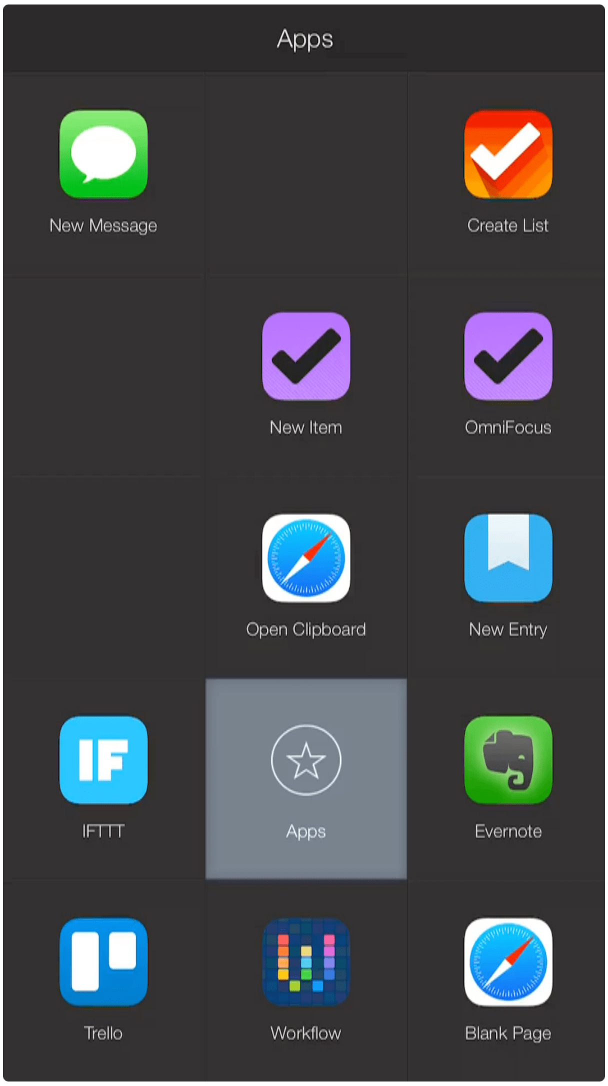
iOS 系统中我们可以通过一段类似 URL 的地址，来启动另外一个应用，并且可以将一些基本参数信息带过去，帮我们完成一件事，简单的说就是 **A 应用呼叫 B 应用，让 B 应用完成一件事，并且可以传递数据给 B 应用**。大家如果不理解也没关系，我举一个简单的例子：

事例 2

我希望在 Launch Center Pro[8] **应用中，输入任务内容和预计完成时间，然后打开 OmniFocus 根据我上一步输入的内容创建任务。**

关联阅读：《效率神器 Launch Center Pro 功能介绍与基础入门》

在这个需求里，最关键的是我们要能在 Launch Center Pro 中启动 OmniFocus，并且告诉它创建一个新任务，以及任务的标题和完成时间。我们看下具体交互是什么样的?


Apps
New Message
Create List
New Item
OmniFocus
Open Clipboard
New Entry
IFTTT
Apps
Evernote
Trello
Workflow
Blank Page

我们看到在 Launch Center Pro 中弹出了两个对话框，分别输入了任务标题和预计完成时间，然后 Launch Center Pro 呼叫并启动了 OmniFocus，同时将用户输入的任务标题和预计完成时间带了过去，这就是 URL Scheme 在起作用。

具体用到的 URL Scheme 代码如下：

```
omnifocus://x-callback-url/add?name=[prompt:Task]&due=[prompt
:Due to…]&x-success={{launch:}}
```

我们看到，URL Scheme 技术成功的使得两个不同的 app 之间，进行了数据互通，A 应用呼叫 B 应用，并且带过去一些数据，两个 app 都能够独立工作，同时也实现了协作。那么另一个非常优秀的移动平台 Android 是如何打破 app 之间的壁垒呢?

Android 平台打破 App 之间壁垒

Android 平台由于系统本身的开放性，它在实现 app 互通上有着很多技术手段，我的所知也有限，这里简单列举两种方式介绍一下：

1. URL Scheme：没错，Android 平台同样支持 URL Scheme，它并不是 iOS 平台独有的，基于 URL Scheme 我们依然可以通过 A 应用呼叫 B 应用，并且传递一些数据过去。
2. Android intent：这是 Android 平台独有的技术，通过在 A 应用内一段代码，实现呼叫 B 应用，同时也可以带一些数据过去。

打破应用和服务之间的壁垒有什么问题

通过从「桌面时代」到「移动互联网时代」的梳理，我们发现技术在升级，平台在进步，各平台都提供了足够丰富的技术实现「打破应用和应用之间的壁垒」，可见这个需求是一个大众需求。但是无论是 PC 平台的脚本技术，还是移动平台的 URL Scheme，都有两个最大的问题：

1. 无法在应用和应用之间传递「复杂」的信息；
2. 在云时代无法实现「跨平台互通」。

为了便于理解以上两个问题，我们再举个例子：

事例 3：批量将 Drafts 笔记导入到 Evernote

通常我会使用 iPhone 上的 Drafts 应用快速记录笔记，一天大概会产生 5–10 条笔记信息，同时我习惯在 Evernote 上管理我所有笔记，这里就产生了一个需求，我希望将 Drafts 上的笔记批量导入到 Evernote 中。

在上例中，批量导入笔记到 Evernote 属于「应用和应用之间传递复杂的信息」，通过上文提到的 URL Scheme 是无法实现的这样复杂信息传递的。同时 Evernote 属于「跨平台应用」[9]，我们最终的目标是将 Drafts 笔记批量导入到 Evernote 中，同时支持在所有平台中查看笔记，这就是典型的「跨平台互通」。

我们知道 Evernote 是可以跨平台查看所有笔记的，那么这是如何实现的呢？在搞清楚之前先明确一个概念：在云计算时代，大部分应用都是在本地收集信息，然后上传到云端进行计算，最终将结果显示给用户。所以，我们可以说：**大部分应用只是服务的表现形式，所以应用的核心是服务，而服务的核心是云计算**。

大家了解了这个概念，我们再推倒一下应用和服务的关系，就能得出如下结论：打破应用和服务之间的壁垒已经不重要了，因为应用即服务，我们更迫切的需求是「打破服务和服务之间的壁垒」。

打破服务和服务之间的壁垒

这里有一个很重要的概念：**服务**。上边我们提到过「大部分应用只是服务的表现形式，它的核心是服务」，那么如何理解**服务**这个概念呢？

还记得刚才的我问题：**Evernote 笔记信息跨平台互通是如何实现的？**

其实很简单，在手机上启动 Evernote 后，它会自动启动同步服务，将笔记上传到云端，当我们在 PC 上访问 Evernote 的时候，Evernote 应用也会启动同步服务，将笔记从云端下载笔记到本地，这样就实现了笔记的同步服务。所以跨平

台互通最核心的还是**同步服务**，那么同步服务又是如何实现的，这里就不得不提到 **API**。

> API（Application Programming Interface 应用服务接口）是一些预先定义的网络服务接口，应用通过 API 实现对服务的操作。如果一个服务的 API 完全开放的话，那么任何其他应用或者服务都可以通过 API 来享受这个服务提供的功能。

大家不用担心，我们不会进行技术细节的探讨，简单说 API 就像是翻译器，通过翻译器我们可以让一个说英语的人和一个说德语的人进行沟通；所以通过 API 我们也可以让 A 服务 和 B 服务沟通，这里举一个大家常见的例子就知道了：

事例 4：通过微信登录到其他 App

在手机上很多 app 都支持通过微信账户直接登录，这样我们只需要记住微信一个账户，就能登录到各个应用，无须再记住密码。这个场景就典型的 API 使用。微信提供了**登录 API**，任何 app 只要请求微信的登录 API ，就可以完成微信登录。

了解了 API 是怎么回事之后，就会发现其实在我们的手机上和 PC 里，充斥着大量的 API，像登录服务，支付服务，搜索服务，通知服务都是 API 的典型应用。只是对于大多数人来说，API 是一个不可察觉的存在。

读到这里，相信大家已经能够理解破除「信息孤岛」是怎么回事以及如何实现的了，这里最核心的就是 API。API 是服务和服务之间的桥梁，有了 API 我们就可以进行「跨平台跨服务协作」了。为了便于大家理解「跨平台服务协作」我再举一个简单的例子：

事例 5：Gmail 邮件稍后读任务

我希望在 Gmail 中如果遇到需要我稍晚一些处理的邮件（比如我在家，但是需要我到公司才处理的邮件）的时候，可以自动在 Todoist 中创建一条提醒任务，提醒我需要稍后处理该邮件。

我们看这个需求里涉及到两个服务之间的协作，分别是 Gmail 和 Todoist，但是这两个服务的场景不同，Gmail 我是在 Mac 上操作，而我希望收到的提醒是在 iPhone 上，macOS 和 iOS 系统是两个平台，这个例子就是一个典型的「跨平台跨服务协作」。

打破信息孤岛带来的便利，从量变到质变

解决信息孤岛带来的最大的便利就是「服务和服务之间的跨平台协作」，这个便利使得人们的生产力获得了极大的提高，通过协作我们可以让多个平台的多个服务为我们所用，让事情自动发生。

我们都知道，一旦事情可以自动化，那么就会发生量变到质变，生产力获得极大提高，大家可以通过事例 6 感受一下：

事例 6：Feedly 稍后读任务

我每天都要在 Feedly 上阅读文章，大部分的文章我都会一扫而过，但是遇到那些需要我深度阅读的文章，我希望能够很方便地将这篇文章保存到 Pocket 中稍后读。

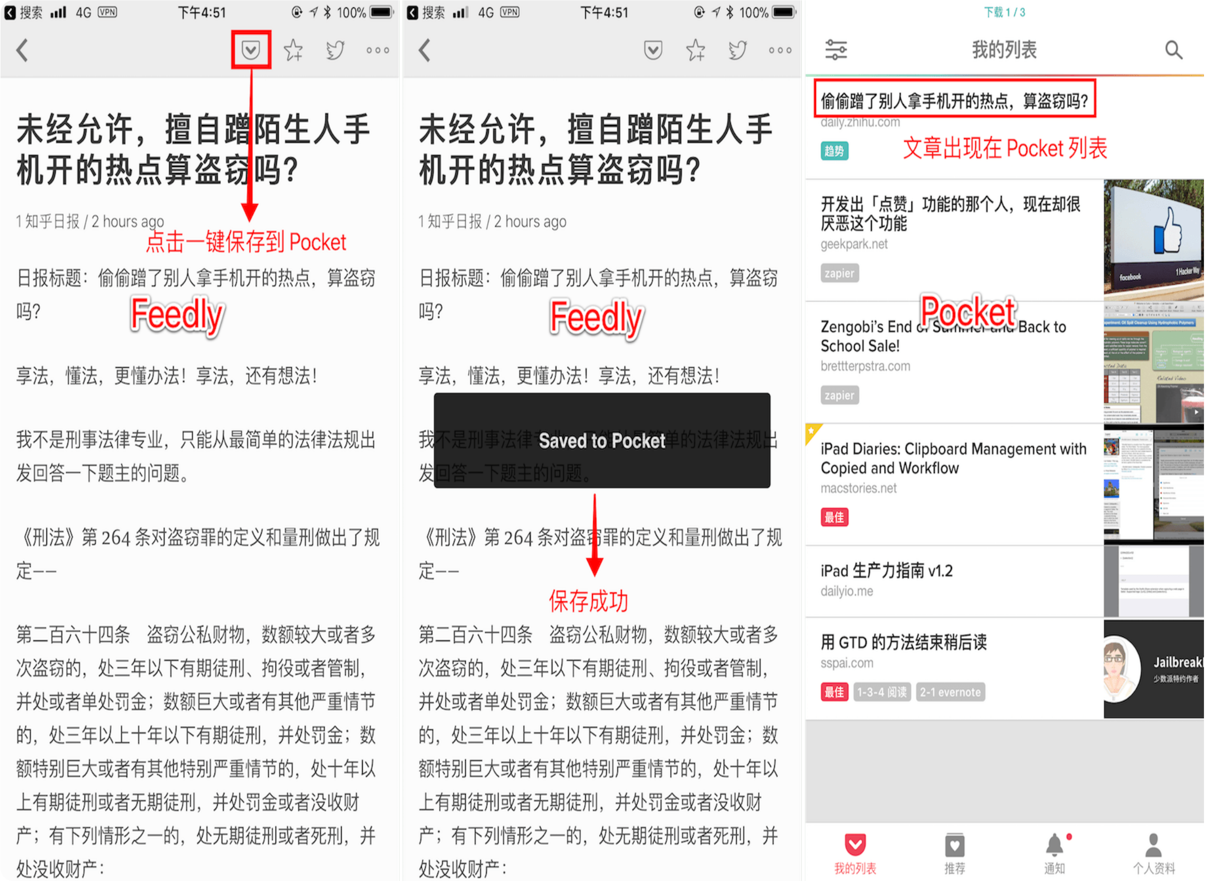
在没有实现自动化的时候，需要以下三步：

1. 遇到需要稍后读的文章，拷贝出文章的 URL；
2. 在浏览器中打开 URL；
3. 选择浏览器的分享功能，分享到 Pocket。

通常我在 Feedly 中阅读文章的时候，采用快速扫读方式，即只看文章标题和摘要。当遇到需要稍后阅读的文章的时，我才会执行以上三步，但是总这样操作

的话，我的流程是经常被打断的，所以效率低，生产力低。但是，如果这个流程能够「自动化」，那么扫读用时就会相映减少，效率提高，生产力自然就高，节省下来的时间我还可以安排更重要的事情去处理。

流程「自动化」的任务，就落在了广大开发者身上，如果一个服务的开发者提供了对另一个服务的集成（当然对方提供了公开 API），那么我们就能享受到足够的便利。在**事例 6**里，涉及到了两个服务 Feedly 和 Pocket，我们的这个需求恰好 Feedly 的开发者想到了，他在应用中提供了此功能，看下图：



未经允许，擅自蹭陌生人手机开的热点算盗窃吗?

1 知乎日报 / 2 hours ago

点击一键保存到 Pocket

日报标题：偷偷蹭了别人拿手机开的热点，算盗窃吗?

Feedly

享法，懂法，更懂办法！享法，还有想法！

我不是刑事法律专业，只能从最简单的法律法规出发回答一下题主的问题。

《刑法》第 264 条对盗窃罪的定义和量刑做出了规定——

第二百六十四条　盗窃公私财物，数额较大或者多次盗窃的，处三年以下有期徒刑、拘役或者管制，并处或者单处罚金；数额巨大或者有其他严重情节的，处三年以上十年以下有期徒刑，并处罚金；数额特别巨大或者有其他特别严重情节的，处十年以上有期徒刑或者无期徒刑，并处罚金或者没收财产；有下列情形之一的，处无期徒刑或者死刑，并处没收财产：

未经允许，擅自蹭陌生人手机开的热点算盗窃吗?

1 知乎日报 / 2 hours ago

日报标题：偷偷蹭了别人拿手机开的热点，算盗窃吗?

Feedly

享法，懂法，更懂办法！享法，还有想法！

我不是刑事法律专业，只能从最简单的法律法规出发回答一下题主的问题。

《刑法》第 264 条对盗窃罪的定义和量刑做出了规定——

第二百六十四条　盗窃公私财物，数额较大或者多次盗窃的，处三年以下有期徒刑、拘役或者管制，并处或者单处罚金；数额巨大或者有其他严重情节的，处三年以上十年以下有期徒刑，并处罚金；数额特别巨大或者有其他特别严重情节的，处十年以上有期徒刑或者无期徒刑，并处罚金或者没收财产；有下列情形之一的，处无期徒刑或者死刑，并处没收财产：

Saved to Pocket

保存成功

偷偷蹭了别人拿手机开的热点，算盗窃吗?
daily.zhihu.com
趋势

文章出现在 Pocket 列表

开发出「点赞」功能的那个人，现在却很厌恶这个功能
geekpark.net
zapier

Pocket

Zengobi's End of Summer and Back to School Sale!
brettterpstra.com
zapier

iPad Diaries: Clipboard Management with Copied and Workflow
macstories.net
最佳

iPad 生产力指南 v1.2
dailyio.me

用 GTD 的方法结束稍后读
sspai.com
最佳 1-3-4 阅读 2-1 evernote

作为开发者能够想到我们的需求并且实现了它，这当然是求之不得的好事，可是难免我们会有个性化的需求，如果没有开发者支持怎么办呢，还记得刚才举的事例 5：

虽然有的邮件客户端提供了邮件任务处理模块，但是我希望我所有跟任务相关的内容全部交给 Todoist 处理，这是一个典型的个性化需求，相信大多数人都

找不到合适的应用或服务提供此功能，而且我们也不可能找 Google 公司（Gmail 的开发商）让他们提供该此功能，那作为普通大众的我们该怎么办呢？

解决类似这个问题就需要我们今天的主角 IFTTT 了。

02. IFTTT 带来了什么

什么是 IFTTT

IFTTT，属于典型的 **If This Then That** 服务，即**如果 A 完成了事情 1，那么就让 B 完成事情 2**，这类服务通常利用互联网上开放的 API，监控用户设置的 **Trigger**，如果 Trigger 被触发则执行用户设置的 **Action**，通常我们可以创建 N 个 **Applet**，来满足我们的各种自动化需求：

- **Trigger**：**触发器**，指用户设置的触发条件，IFTTT 会以轮询的方式监控用户设置的触发条件是否达成；

- **Action**：**动作**，指用户设置好的一系列动作内容，当 IFTTT 发现触发条件达成，就会继续执行用户设置好的动作；
- **Applet**： 我们设置了一个 Trigger 和一个 Action 来实现业务的自动化，这样的一套组合我们称之为一个 Applet。

看到这里如果还不能理解 IFTTT 是什么也没关系，还记得**事例 5**吗，我们加上触发器和动作重新解释一下事例 5：

> 我希望在 Gmail 中如果遇到需要我稍晚一些处理的邮件（比如我在家，但是需要我到公司才处理的邮件）的时候，点击 Gmail 的加星按钮，就可以自动在 Todoist 中创建一条提醒任务，提醒我需要稍后处理该邮件。

这个需求，我们用 **if This Then That** 套一下事例 5 就变成了以下内容：

> 如果在 Gmail 中对一封邮件进行了加星标处理，那么自动到 Todoist 中创建一条稍后处理邮件的任务。

这里的 Trigger 就是「如果在 Gmail 中对一封邮件进行了加星标处理」，对应的 Action 就是 「在 Todoist 中创建一条提醒任务」，有了触发器和动作我们就可以设置一条 Applet 了。

IFTTT 的特点

IFTTT 因其整合了众多互联网服务的 API，使得 IFTTT 成为了自动化服务的先驱和领导者，IFTTT 巧妙的通过 **if This Then That** 将服务与服务串联起来，从而实现了自动化，我们看一下 IFTTT 有哪些特点呢?

- IFTTT 支持 2 步规则，即如果 A 服务发生了事情 1，那么就让 B 服务发生事情 2；
- IFTTT 中创建自动化服务非常方便，设置好触发器和对应的动作即可，学习成本很低；
- IFTTT 中集合了大量的用户分享的 Applets，基本上涵盖了主流的服务商，其他的模仿者在数量上还无法企及。

如何找到自动化实例

IFTTT 提供了许多途径让我们找到合适的实例：

- 官方提供的它们最喜欢的自动化实例，点击访问
- 官方制作的自动化实例合集，点击访问
- 官网或 app 提供的探索功能，点击访问
- 官网或者 app 提供的搜索功能，可按类型筛选，点击访问

03. 国内为什么没有 IFTTT 服务

WECHAT
ALIPAY
WEIBO
BAIDU
INTERNET
NO IFTTT

其实早在 IFTTT 火遍美国硅谷的时候，国内就有了第一家模仿者「如果云」，后来如雨后春笋般涌现了大批模仿者，我们看看他们名字：

- 如果是
- 如果就
- 如果说
- 假设就
- 一旦就

可惜，以上接地气的「模仿者」，没有一家活到现在，究竟是什么原因造成这个现象呢？

1. 新浪微博、腾讯、豆瓣、百度、阿里巴巴等都提供了良好定义的 API，但是即使这些大厂出品的 API，依然不稳定。

   就拿新浪微博 API 举例，在新浪微博推出 API 后，app 市场上涌现了大量优秀的第三方微博客户端，受欢迎程度甚至超过了官方，但是随着新浪微博收紧 API 的权限，许多功能都不再提供，许多特性不再支持，导致大量的第三方 app 停止更新或者下架。试想一下，这样的大厂出品都能随意修改权限和逻辑，何来的稳定性呢？

设权限和逻辑，何来的稳定性呢？

2. 国内互联网环境被几个大的派系把持，腾讯系、阿里系、百度系等，各派系都有直接或者间接的竞争对手，出于商业保护和竞争目的，很难想象他们能够提供功能齐全的 API。

3. 在互联网大厂无法提供更加开放的 API 情况下，有可能提供优质 API 的独立开发者也同样不给力。

   一方面他们的应用或服务用户太少，不具备普遍性；另一方面，如果他们做大了，就会面临被大厂收购的命运，如果被收购进了「派系」，结果就可想而知了。

4. 对于 IFTTT 这样的服务，必须要求用户登陆各种服务账户，有的甚至是将用户名和密码交给平台托管，在国内的安全环境下，用户很难放心的将账户信息交给平台托管，这也是模仿者无法继续的原因之一。

纵观这些模仿者的兴起和衰败可以发现，在国内互联网的土壤上，短时间内还无法生长长出类似 IFTTT 这样的参天大树。

04. 总结

本篇文章从「应用和服务的信息孤岛现象」引出了「打破服务和服务之间的壁垒」，然后又讲到 IFTTT 如何帮我们实现「自动化业务」，最后讲到为什么国内没有出现类 IFTTT 服务，相信大家已经对 IFTTT 不再那么陌生了。

IFTTT 作为典型的 **If This Then That** 服务，效率极高，门槛也相对较低，如果你对自动化有需求，想了解最先进的互联网服务业务，可以从本篇入手。在后续文章中，我们会陆续介绍如何创建自动化业务，以及一些高级玩法和进阶玩法，欢迎大家继续阅读。

1. AppleScript是苹果公司开发的一种脚本语言，可以用来控制。↩
2. 批处理（Batch），也称为批处理脚本。顾名思义，批处理就是对某对象进行批量的处理，通常被认为是一种简化的脚本语言，它应用于 DOS 和 Windows 系统中。↩
3. 超文本标记语言，标准通用标记语言下的一个应用。↩
4. 可扩展标记语言，标准通用标记语言的子集，是一种用于标记电子文件使其具有结构性的标记语言。↩
5. 逗号分隔值（Comma-Separated Values，CSV，有时也称为字符分隔值，因为分隔字符也可以不是逗号），其文件以纯文本形式存储表格数据（数字和文本）。↩
6. JSON（JavaScript Object Notation，JS 对象标记）是一种轻量级的数据交换格式。它基于 ECMAScript（w3c 制定的 JS 规范）的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。↩
7. @JailbreakHum 在 Checked 节目中提到过前云时代这个概念，特指没有云计算的时代，大部分应用都在本地进行计算和服务用户。↩
8. Launch Center Pro，iOS 系统中优秀的效率类软件，多次入选 AppStore 年度精选，一经推出，广受用户好评，通过它用户可以很方便的启动应用，打开网页，甚至能够直接开启应用内的某一功能。↩
9. 如果一个应用可以在多个操作系统和硬件平台下使用，那么这个应用就属于「跨平台应用」。↩

♡

♡ 24

上一期
序言 | 玩转 IFTTT，开启智能生活

下一期
答疑汇总 01